SITEKIN: Jurnal Sains Teknologi dan Industri, Vol. 20, No. 2, June 2023, pp.752 - 763
ISSN 2407-0939 print/ISSN 2721-2041 online

A Web Service Model for Signing Documents Using Certificate-Based Digital Signature

Yusra[1], Muhammad Fikry[2]
[1.2] Department of Informatics Engineering, Faculty of Science and Technology, UIN Sultan Syarif Kasim Riau
Jl. HR. Soebrantas No. 155 Simpang Baru, Panam, Pekanbaru, 28293
Email: yusra@uin-suska.ac.id, muhammad.fikry@uin-suska.ac.id*

ABSTRACT

Since the Covid-19 pandemic started, seminar documents at the Department of Informatics Engineering, UIN Sultan Syarif Kasim Riau, have switched from papers to PDF files. To create a certificate-based digital signature, lecturers use desktop-based PDF processing applications or an Android application developed in the previous study. Unfortunately, lecturers who own iPhone does not have an application with similar functionality. The problem is that the Swift programming language for iOS does not support signing PDF files with digital certificates, and third-party libraries are costly. Therefore, this research develops a digital signature system model with web services for signing on the server side. The results of black box testing on web services and its mobile client application meets the requirements stated in the software requirement specifications. The User Acceptance Test results show that the digital signature system meets the user's needs.

Keywords: *digital signatures, digital documents, mobile applications, web services*

Introduction

Since the COVID-19 pandemic hit Indonesia in March 2020, documents for practical work and final project seminars at the Department of Informatics Engineering, Sultan Syarif Kasim Riau State Islamic University, have changed from paper to digital. The digital document format used is Portable Document Format (PDF). This format supports using a certificate-based digital signature to authenticate documents [1]. A digital ID is required to sign a PDF file, which contains a private key and a certificate with a public key, and other data related to a person or entity [2].

Softwares used by lecturers to create a certificate-based digital signature are desktop-based PDF processing applications such as Adobe Acrobat, and an Android-based mobile application named Tanda Tangan Digital (TTD) Suska which was developed in the previous study [3]. Unfortunately, lecturers who own iPhones, or as many as 22% of all lecturers in the Department of Informatics Engineering, do not have an application with similar functionality. Based on search results on Apple's App Store, most signing applications are image-based, not certificate-based, including Adobe Acrobat Reader. Certificate-based signing applications such as Privy is paid per signature.

To support digital transformation in the Department of Informatics Engineering, an iOS-based mobile application for digital signatures is needed. However, the PDFKit framework [4] on the Swift programming language does not support certificate-based digital signing, and the third-party library PSPDFKit [5] is costly. Gamalielsson et al. [6] suggest using open-source libraries for digital signing and validation of PDF files that are at least as effective as Adobe Acrobat, namely PDFBox [7] and iText [8]. Both use the Java programming language, so they cannot run on the iOS platform, which does not have the Java Runtime Environment.

The problem of platform differences can be solved by using a web service. Web services enable different software platforms and frameworks to work together [9]. According to Schermann et al. [10], as cited in Neumann et al. [11], the architecture that has become the standard (de facto) way of presenting web services is Representational State Transfer (REST). REST is a stateless client-server architecture where web services are used as resources and accessed through their respective Uniform Resource Identifiers (URIs) [12]. Client applications use HTTP commands such as GET, POST, PUT, and DELETE to access REST services. Clients can be mobile applications [12]–[17], and web-based applications [18], [19].

Based on the search results regarding digital signing, only one study used REST architecture. Ribeiro et al. [19] developed a digital signature system model using the XML Advanced Electronic Signature (XadES) Standard, as a REST service, for use at the University of Brasilia. The digital document format is XML, not PDF, and the client application is web-based, not mobile. In addition, there are studies developing desktop-based [20] [21] and web-based [22] [23] solutions to sign PDF files with an invisible signature. There are also

studies developing web-based solutions [24]–[27], Android [28] [29], and Telegram [30] to sign PDF files with a visible signature.

This study created a digital signature system model with a client-server architecture. The system provides REST services that enable digital signing and validation of PDF files on the server side. The mobile application acts as a client requesting the addition of a digital signature to a document and requesting validation of the signature. Users can determine the look and location of the visible signature on a PDF page.

Research Methods

Figure 1 shows the stages of this study. In the first stage, we collected a digital ID file, digital certificate files from lecturers, and PDF files of seminar documents.

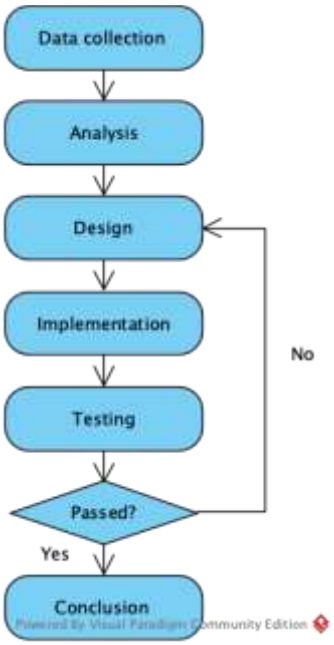

Figure 1. Research stages

The Unified Modeling Language (UML) is used in the next stage to model the proposed system architecture. Based on the analysis and design results, web services were implemented in the PHP programming language, involving the open-source PDFBox library, as well as an iOS application in the Swift programming language. At the testing stage, black-box test and User Acceptance Test are carried out. If the test results do not meet the requirements specifications, revisions are made. In the final stage, conclusions are drawn.

Results and Discussion

The digital signature system consists of two subsystems: web services on the server side and mobile applications on the client side. Software requirements specifications are based on previous research [3] with the addition of web services, as follows :

1. The lecturer signs in to the mobile application using an Apple ID. The web service creates a new account based on Apple ID.
2. The lecturer uploads a digital ID. The web service creates a signature preview based on the default settings and the information contained in the digital ID.
3. The lecturer enters the digital ID password.
4. The lecturer determines the look of the certificate-based signature by specifying the type of signature, image, description, and image background. The web service creates a preview of what the signature looks like based on these settings and the information contained in the digital ID.
5. The lecturer opens a PDF file.
6. The lecturer writes a text on a PDF page. The web service renders text on the PDF file.
7. The lecturer makes a signature in the desired size and position on a PDF page. The web service adds a signature on the PDF file.

8. The lecturer asks for validation of signatures found in the PDF file. The web service compares it against a list of trusted digital certificates and marks whether it is registered or not, and valid or not. The web service returns signature details.
9. The lecturer saves the PDF file on storage media.
10. The lecturer shares the PDF file with other applications.
11. The lecturer signs out of the application. The web service deletes the lecturer's account. This functionality is used before the lecturer uninstalls the mobile application.

UML diagrams, created using Visual Paradigm, are used to model the system architecture. The use case diagram in Figure 2 shows the system's association between actor and use cases.

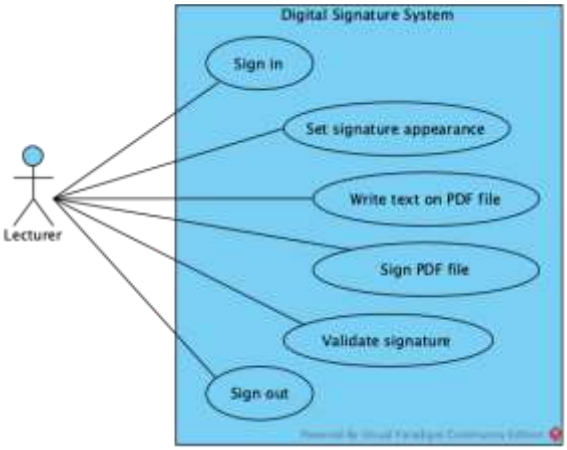

Figure 2. Use case diagram

The activity diagram in Figure 3 shows a sequence of steps when the lecturer uses the mobile application, from signing in to sharing a signed PDF file.

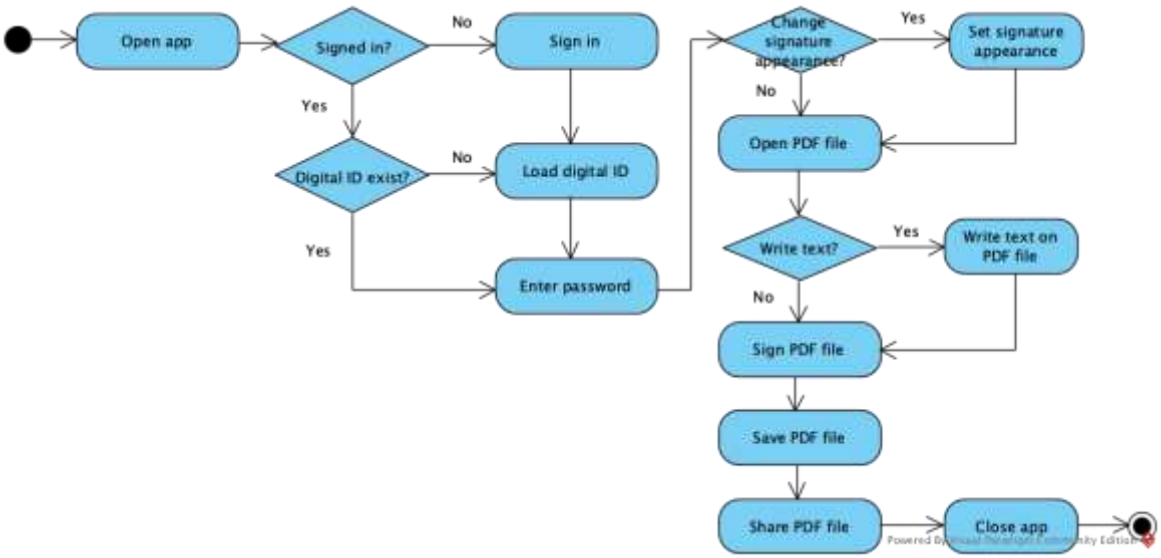

Figure 3. Activity diagram

The package diagram in Figure 4 shows a digital signature system partitioned into two subsystems, denoted as packages. Packages are used to group model elements.

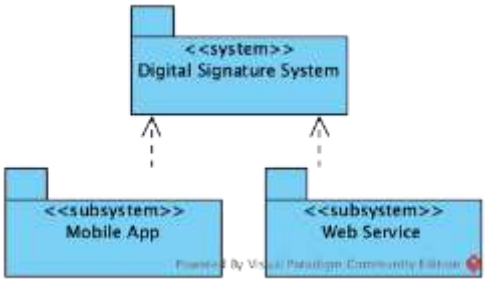

Figure 4. Package diagram

A class diagram is used to show the internal architecture of a system. Figure 5 shows classes in the Mobile App subsystem package.

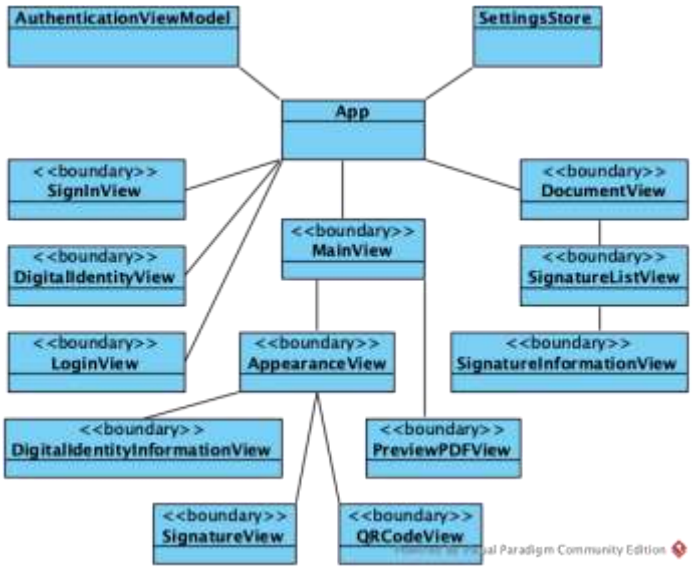

Figure 5. Class diagram

Based on software functionalities, the Web Service subsystem provides seven services. First, the service for creating a new account based on Apple ID is shown in Table 1.

Table 1. Sign-in service

URL	https://site/path/signin
Method	POST
URI parameter	token

Success response	{
	"status": 200,
	"success": true,
	"message": "You have successfully signed in"
	}

Second, the service uploads the digital ID and then creates a preview of the signature based on the default settings and information on the digital ID. The service specifications is shown in Table 2.

Table 2. Save service

URL	https://site/path/save
Method	POST
URI parameter	account, password
File	digital ID
Success response	{

	"status": 200, "success": true, "message": "" }

Third, the service makes a signature preview based on the settings and information on the digital ID. The service specification is shown in Table 3.

Table 3. Signature preview service

URL	https://site/path/preview
Method	POST
URI parameter	account, password, input, output, page_index, appearance, show_label, show_email, show_data, show_reason, show_location, description_label, description_date, description_reason, description_location, show_background, x, y, width, height
File	digital ID, image (optional), background image (optional)

Success response	{
	"status": 200,
	"success": true,
	"message": "",
	"data": "URL for PDF"
	}

Fourth, the service for writing text in PDF files. The text location on a PDF page is determined by the x, y, width, and height parameters. The service specification is shown in Table 4.

Table 4. Write service

URL	https://site/path/write
Method	POST
URI parameter	account, password, input, output, page_index, description_label, x, y, width, height
File	PDF document

Success response	{
	"status": 200,
	"success": true,
	"message": "",
	"data": "URL for PDF"
	}

Fifth, the service adds a digital signature to a PDF file. The size and position of a signature on the PDF page is determined by the x, y, width, and height parameters. The service specification is shown in Table 5.

Table 5. Sign service

URL	https://site/path/sign
Method	POST
URI parameter	account, password, input, output, page_index, appearance, show_label, show_email, show_data, show_reason, show_location, description_label, description_date, description_reason, description_location, show_background, x, y, width, height
File	PDF document
Success response	{

```
"status": 200,
"success": true,
"message": "",
"data": "URL for PDF"
}
```


Sixth, the service validates signatures in a PDF file against a list of trusted digital certificates. The data attribute in the success response is an array containing zero or more signature information objects. The service specification is shown in Table 6.

Table 6. Signature validation service

URL	https://site/path/validate
Method	POST
URI parameter	account
File	PDF document

Success response	{
	"status": 200,
	"success": true,
	"message": "",
	"data": [
	{
	"no":0,
	"tanda_tangan": "",
	"oleh": "",
	"email": "",
	"pada": "",
	"alasan": "",
	"lokasi": "",
	"field": "",
	"halaman": 0,
	"revisi": 0,
	"autentik": "",
	"verifikasi": "",
	"validitas_ketika_menandatangani": "",
	"validitas_hari_ini": "",
	"perubahan_dokumen": "",
	"signature_algorithm": "",
	"filter_type": "",
	"filter_subtype": "",
	"serial_number": "",
	"jenis": "",
	"self_signed": "",
	"nama_penerbit": "",
	"email_penerbit": "",
	"organisasi_penerbit": "",
	}, …]
	}

Seventh, the service for account deletion when a user signs out is shown in Table 7.

Table 7. Sign out service

URL	https://site/path/signout
Method	POST
URI parameter	account
Success response	{ "status": 200, "success": true, "message": ""

```
}
```

MySQL database is used to store account data. An account is created when the user signs in and deleted when the user signs out. Figure 6 shows the main screen layout, and Figure 7 shows the signature's settings screen layout.

Figure 6. Main screen layout

Figure 7. Settings screen design

Figure 8 shows a deployment diagram showing the physical architecture of the system. This diagram shows the relationship between hardware and software in the system.

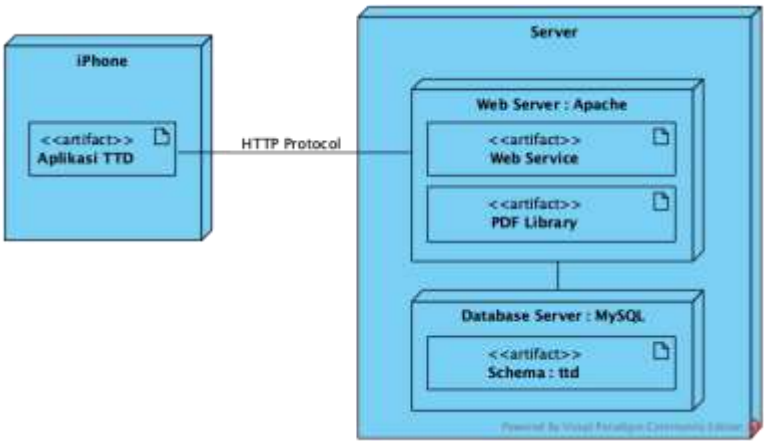

Figure 8. Deployment diagrams

Based on the analysis and design results, web services were implemented in the PHP programming language, an application for digital signature in the Java programming language involving the PDFBox open source library, as well as an iOS application in the Swift programming language. The web service is hosted on the department's server.

On the main screen (Figure 9), click the "Rupa tanda tangan" button to open the settings screen (Figure 10). On the settings screen, the user can determine the look of the certificate-based signature by specifying the type of signature, image, description, and background image. Images can be taken from the Photos app, Files app, a custom-made signature, or based on a QR Code.

Figure 9. Main screen

Figure 10. Settings screen

Figure 11 shows a PDF page with a signature rectangle (white box with red border) that can be resized and dragged where the signature appears, and Figure 12 shows the page after signing.

Figure 11. Before being signed

Figure 12. After being signed

Figure 13 shows the validation results of each signature as a green-colored (REGISTERED) label or a red-colored (UNREGISTERED) label. Each signature can be selected to get detailed information.

Figure 13. Validation screen

Figure 14 shows detailed information about a signature.

Figure 14. Signature screen

Signed PDF files can be opened in Adobe Acrobat, as shown in Figure 15.

Figure 15. A digital signature (marked with a red box on the right) and its detail (on the left)

Based on the black-box test result, web services and its mobile client application are in accordance with the results of the previous stages. User Acceptance Test (UAT) was conducted on 3 iPhone users (lecturers). The questionnaire consists of 8 questions with a 5-point Likert scale answer format. Based on the UAT result, all respondents stated that the digital signature system was helpful in their work, easy to use, and the results were as expected. All respondents also strongly agreed that the signature validation functionality was helpful.

The client application is published on App Store and can be installed on iPhone or iPad with iOS version 14 or higher. Figure 16 shows the application landing page on App Store. This application is intended for lecturers in the Department of Informatics Engineering, State Islamic University of Sultan Syarif Kasim Riau.

Figure 16. Application landing page on App Store

Conclusion

A digital signature system with web services and its client application has been successfully developed, tested, and published. Based on the test results, it is concluded that web services and their client application meet the software requirement specifications and also meet the user's needs.

The mobile client application for Android and iOS requires a minimum version of a supported operating system. To support devices that do not meet that requirement, it is necessary to develop a web-based client application for signing PDF files. One of the challenges is how users can interact with PDF files on a web browser to determine the position of the signature placement. In the future, new functionalities can be added to client applications, such as signing at a specified signature field and creating a list of trusted signatures.

Bibliography

[1] Adobe Systems Incorporated, *Document management - Portable document format - Part 1: PDF 1.7 (ISO 32000-1:2008)*, 1st ed. 2008.

[2] J. Chandrashekhara, "A Comprehensive Study on Digital Signature," *Int. J. Innov. Res. Comput. Sci. Technol.*, vol. 9, no. 3, pp. 43–47, 2021, doi: 10.21276/ijircst.2021.9.3.7.

[3] M. Fikry and Yusra, "Aplikasi Android untuk Tanda Tangan Digital (Studi Kasus di Jurusan Teknik Informatika, UIN Suska Riau)," *SITEKIN J. Sains, Teknol. dan Ind.*, vol. 19, no. 2, pp. 430–435, 2022.

[4] "PDFKit," 2022. https://developer.apple.com/documentation/pdfkit

[5] "PSPDFKit," 2022. https://pspdfkit.com

[6] J. Gamalielsson, F. Jakobsson, B. Lundell, J. Feist, T. Gustavsson, and F. Landqvist, "On the Availability and Effectiveness of Open Source Software for Digital Signing of PDF Documents," in *IFIP International Conference on Open Source Systems*, 2015, vol. 451, pp. 71–80. doi: 10.1007/978-3-319-17837-0.

[7] "Apache PDFBox," 2022. https://pdfbox.apache.org

[8] "iText," 2022. https://itextpdf.com

[9] W3C Working Group, "Web Services Architecture," *2004*. https://www.w3.org/TR/ws-arch/

[10] G. Schermann, J. Cito, and P. Leitner, "All the Services Large and Micro: Revisiting Industrial Practice in Services Computing," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9586, 2016, pp. 36–47. doi: 10.1007/978-3-662-50539-7_4.

[11] A. Neumann, N. Laranjeiro, and J. Bernardino, "An Analysis of Public REST Web Service APIs," *IEEE Trans. Serv. Comput.*, vol. 14, no. 4, pp. 957–970, 2021, doi: 10.1109/TSC.2018.2847344.

[12] B. D. Pratomo and K. Haryono, "Perancangan RESTful Web Service Satuan Kredit Partisipasi di Universitas Islam Indonesia," *Semin. Nas. Din. Inform.*, pp. 74–77, 2020.
[13] R. S. Galih and F. Salamun, "Implementasi Web Service pada Aplikasi Mobile untuk Mendukung Sistem Informasi di Bandung N-Max Community," in *Konferensi Nasional Sistem Informasi*, 2018.
[14] D. Darmawan, F. S. F. Kusumah, and S. H. Al Ikhsan, "Web Service untuk Transaksi Data pada Aplikasi Fasilitas Keuangan dengan Metode REST," *J. Sains Komput. Inform.*, vol. 5, no. 2, pp. 852–865, 2021.
[15] M. Nuraminudin, "Implementasi Teknik Hybrid Mobile Application dalam Pembuatan Aplikasi Mobile Marketplace Ikan Hias," *INFOS J. - Inf. Syst. J.*, vol. 2, no. 1, pp. 7–12, 2019, [Online]. Available: https://ojs.amikom.ac.id/index.php/INFOSJournal/article/view/2422/2242
[16] R. Choirudin and A. Adil, "Implementasi REST API Web Service dalam Membangun Aplikasi Multiplatform untuk Usaha Jasa," *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 18, no. 2, pp. 284–293, 2019, doi: 10.30812/matrik.v18i2.407.
[17] D. I. Pradana and I. Waspada, "Aplikasi Hybrid pada Sistem Informasi Penyewaan Buku," *Simetris J. Tek. Mesin, Elektro dan Ilmu Komput.*, vol. 10, no. 1, pp. 1–14, 2019, doi: 10.24176/simet.v10i1.2600.
[18] O. D. Arianto and Y. A. Susetyo, "Penerapan RESTful Web Service dengan Framework Laravel untuk Pembangunan Sistem Informasi Manajemen Sumber Daya Manusia," *JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 7, no. 2, pp. 522–532, 2022, doi: 10.29100/jipi.v7i2.2870.
[19] R. C. Ribeiro, M. G. De Almeida, and E. D. Canedo, "A digital signature model using XAdES standard as a REST service," *Inf.*, vol. 12, no. 8, 2021, doi: 10.3390/info12080289.
[20] A.-H. Anastacio, F.-M. Heberto, T.-M. Cristhian, and O.-R. Juan Carlos, "Management of Digital Documents with Encrypted Signature, Through the Use of Centralized PKI, and Distributed Using Blockchain for a Secure Exchange," *J. Res. Dev.*, vol. 5, no. 15, pp. 26–37, 2019, doi: 10.35429/jrd.2019.15.5.26.37.
[21] Y. Suharya and H. Widia, "Implementasi Digital Signature Menggunakan Algoritma Kriftografi RSA untuk Pengamanan Data di SMK Wirakarya 1 Ciparay," *J. Inform.*, vol. 7, no. 1, pp. 20–29, 2020.
[22] Sugiyatno and P. D. Atika, "Digital Signature dengan Algoritma SHA-1 dan RSA Sebagai Autentikasi," *J. Cendikia*, vol. 16, no. 2, pp. 74–83, 2018.
[23] N. Arwa, Aminudin, and S. Arifianto, "Implementasi Tanda Tangan Digital menggunakan ECDSA (Studi Kasus: Jurnal Tipe File PDF)," *J. Repos.*, vol. 3, no. 3, pp. 321–330, 2021, doi: 10.22219/repositor.v2i3.1306.
[24] W. Sholihah, S. Indriasari, I. Noviyanti, A. Mardiyono, and N. Aziezah, "ESVISIGN: Tanda Tangan Digital Sekolah Vokasi IPB," *J. Teknol. Inf. dan Multimed.*, vol. 3, no. 4, pp. 217–226, 2022, doi: 10.35746/jtim.v3i4.188.
[25] T. Abdurrachman and B. R. Suteja, "Pengembangan Sistem Informasi Asosiasi Jasa Konstruksi dengan Menerapkan Tanda Tangan Digital," *J. Tek. Inform. dan Sist. Inf.*, vol. 7, no. 1, pp. 261–273, 2021, doi: 10.28932/jutisi.v7i1.3431.
[26] H. Indriyawati, T. Winarti, and V. Vydia, "Web-based Document Certification System with Advanced Encryption Standard Digital Signature," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 22, no. 1, pp. 516–521, 2021, doi: 10.11591/ijeecs.v22.i1.pp516-521.
[27] A. El Mane, Y. Chihab, and R. Korchiyne, "Digital Signature for Data and Documents using Operating PKI Certificates," in *SHS Web of Conferences*, 2021, vol. 119. doi: 10.1051/shsconf/202111907004.
[28] I. G. Firmansyah and R. B. Hadiprakoso, "Rancang Bangun Aplikasi PDF Signer Berbasis Android pada PDAM Kabupaten Tuban," *J. Ilm. Ilmu Komput.*, vol. 7, no. 2, pp. 57–61, 2021, doi: 10.35329/jiik.v7i2.202.
[29] R. A. Perdana, D. R. Anbiya, and A. Grahitandaru, "Penerapan Tanda Tangan Digital pada Gambar Formulir C1.Plano-KWK di Pilkada Sulawesi Selatan," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 6, no. 5, pp. 475–484, 2019, doi: 10.25126/jtiik.201961471.
[30] H. Kabetta, "Desain dan Implementasi Penandatangan Elektronik Sertifikat X509 Menggunakan Platform Bot Telegram," *Telematika*, vol. 13, no. 1, pp. 22–35, 2020, doi: 10.35671/telematika.v13i1.936.